

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[First Hit](#)



Generate Collection

L2: Entry 3 of 3

File: JPAB

Sep 8, 2000

DOCUMENT-IDENTIFIER: JP 2000242461 A

TITLE: IMAGE PROCESSOR

Abstract Text (2):

SOLUTION: A command separating part 11 classifies a print description language(PDL) inputted to an input terminal A into commands corresponding to a multilevel character, binary character, multilevel vector, binary vector, graphics and raster image. A binary vector processing part 15 converts the vector command into a binary edge list. A graphics processing part 16 converts the graphics plotting command into a multilevel edge list. A raster image processing part 17 converts the raster image plotting command into a multilevel edge list. A composing part 18 composes the edge lists from the respective processing parts, converts the composite list into the raster image, outputs it to an output terminal A, generates classification data showing the classification of respective pixels in the raster image and outputs them to an output terminal B.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-242461
(P2000-242461A)

(43) 公開日 平成12年9月8日(2000.9.8)

| (51) Int.Cl. ⁷ | 識別記号 | F I | テーマコード*(参考) |
|--------------------------------------|------|--------------|-------------------|
| G 0 6 F 3/12 | | G 0 6 F 3/12 | G 2 C 0 6 2 |
| | | | L 2 C 2 6 2 |
| B 4 1 J 2/52 | | G 0 9 G 5/00 | 5 2 0 W 5 B 0 2 1 |
| 2/485 | | 5/40 | 5 B 0 8 0 |
| G 0 6 T 11/00 | | B 4 1 J 3/00 | A 5 C 0 8 2 |
| 審査請求 未請求 請求項の数 8 O L (全 22 頁) 最終頁に続く | | | |

(21) 出願番号 特願平11-42359

(22) 出願日 平成11年2月19日(1999.2.19)

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72) 発明者 河野 裕之

神奈川県海老名市本郷2274番地 富士ゼロックス株式会社内

(72) 発明者 月村 滋

神奈川県海老名市本郷2274番地 富士ゼロックス株式会社内

(74) 代理人 100098084

弁理士 川▲崎▼ 研二

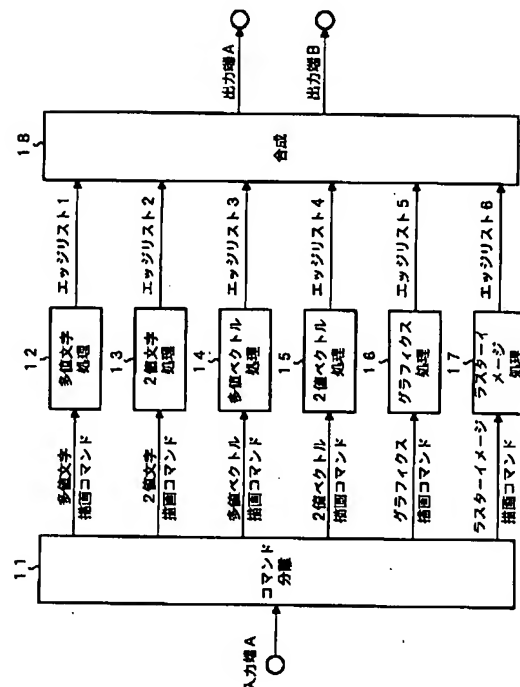
最終頁に続く

(54) 【発明の名称】 画像処理装置

(57) 【要約】

【課題】 ページ記述言語で記述された画像情報をラスター画像に変換する際に必要な記憶容量を少なくし、より高画質のプリントアウトを得る。

【解決手段】 コマンド分離部11は、入力端Aに入力されたPDLを、多値文字、2値文字、多値ベクトル、2値ベクトル、グラフィクス、ラスターイメージに対応するコマンドに分類する。2値ベクトル処理部15は、ベクトルコマンドを2値エッジリストに変換する。グラフィクス処理部16は、グラフィクス描画コマンドを多値エッジリストに変換する。ラスターイメージ処理部17は、ラスターイメージ描画コマンドを多値エッジリストに変換する。合成部18は、各処理部からのエッジリストを合成するとともに、ラスター画像に変換して出力端Aに出力するとともに、ラスター画像の各画面の分類を示す分類データを生成して出力端Bに出力する。



【特許請求の範囲】

【請求項1】 ページ記述言語で記述された画像情報を
入力する入力手段と、

前記入力手段で入力された前記画像情報の中から文字ま
たは線画部を抽出する抽出手段と、

前記抽出手段で抽出した文字または線画部を、ラスタ
ー展開するための中間的なコードであり、2値の階調を有
する第1の中間コードに変換する第1の変換手段と、

前記第1の変換手段により変換された第1の中間コード
を、当該第1の中間コードより低い解像度で多値の階調
を有する第2の中間コードに変換する第2の変換手段
と、

前記第1の変換手段により変換された第1の中間コード
または前記第2の変換手段により変換された第2の中間
コードに基づいて、前記入力手段で入力された前記画像
情報の所定の領域毎に、当該領域を占める画像の種類を
表すタグ情報を生成するタグ生成手段とを具備し、
前記タグ生成手段が生成する前記タグ情報は、前記第2
の中間コードで表された文字または線画部であることを
表す情報を少なくとも含むことを特徴とする画像処理装
置。

【請求項2】 前記第2の中間コードと、当該第2の中
間コードとは異なる中間コードとを合成する合成手段を
有することを特徴とする請求項第1項に記載の画像処理
装置。

【請求項3】 前記文字または線画部と、前記文字また
は線画部以外の部分との境界を検知する境界検知手段を
有し、

前記タグ生成手段が生成する前記タグ情報は、前記境界
検知手段により検知された境界であることを表す情報を
さらに含むことを特徴とする請求項第2項に記載の画像
処理装置。

【請求項4】 前記文字または線画部と、前記文字また
は線画部以外の部分との境界を検知する境界検知手段を
有し、

前記境界検知手段における検知結果により前記合成手段
における合成を異ならせることを特徴とする請求項第2
項に記載の画像処理装置。

【請求項5】 前記第2の中間コードをラスタ
ー展開した画像と、前記第2の中間コードとは異なる中間コード
をラスタ
ー展開した画像とを合成する合成手段を有する
ことを特徴とする請求項第1項に記載の画像処理装置。

【請求項6】 前記文字または線画部と、前記文字また
は線画部以外の部分との境界を検知する境界検知手段を
有し、

前記タグ生成手段が生成する前記タグ情報は、前記境界
検知手段により検知された境界であることを表す情報を
さらに含むことを特徴とする請求項第5項に記載の画像
処理装置。

【請求項7】 前記文字または線画部と、前記文字また

は線画部以外の部分との境界を検知する境界検知手段を
有し、

前記境界検知手段における検知結果により前記合成手段
における合成を異ならせることを特徴とする請求項第5
項に記載の画像処理装置。

【請求項8】 前記タグ情報は、前記第2の中間コード
で表された文字または線画部の階調数により内容を異なら
せることを特徴とする請求項第1ないし7のいずれか
に記載の画像処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、画像処理装置に
関し、特に文字・線画を表す文字・線画データをラスタ
ー画像データに変換して出力する画像処理装置に関する。

【0002】

【従来の技術】近年、パーソナルコンピュータ（以下、
PCという）やワークステーション（以下、WSとい
う）で作成された文書をプリントアウトする機会が増え
ている。特に、オフィスでプリントアウトされる文書
は、文字・グラフィクスあるいはスキャナで読み込んだ
写真等が同一ページ内にレイアウトされたものが多く、
最近では様々な色で彩られたカラー文書もプリントア
ウトする機会が増えている。また、複数のPCやWSがネ
ットワークを介してプリンタを共有するシステム構成が
とられることも多く、プリンタに対する高速化・高画質
化の要求はますます強くなっている。

【0003】ところで、PCやWSからプリンタへのデ
ータ転送時間を短縮するためと、特性の異なるプリンタ
でもできるだけ同一の内容でプリントアウトするため
に、文書を描画コマンドの集合体であるプリント記述言
語（以下、PDLという）で記述してプリンタに転送
し、プリンタでPDLを解釈してラスタ
ー画像に変換してプリントアウトすることが一般的となっ
ている。しかしながら、現存するプリンタの解像度に合
せてPDLをラスタ
ー画像に変換してプリントアウトすると、特に文
字や線画のエッジにジャギーを生じてしまう。

【0004】そこで、上述したジャギーを低減するため
に、例えば米国特許第4,847,641号に開示され
ている技術のように、変換後のラスタ
ー画像からジャ
ギーが発生しうる領域をパターンマッチングによって抽
出して補正を施す方法が提案されている。しかしながら、
この方法では、例えば明朝体文字のトメやハネのよう
に、補正を施すべきでない領域であっても補正されて
しまい、字体の変形を引き起こしてしまう。一方、太
さが均一で等間隔かつ平行な直線群が解像度の制限に
よってラスタ
ー画像への変換の際に、太さにばらつきが生じ
たり、あるいは等間隔でない平行な直線群に変換され
るような場合には、上記従来技術では補正できない。

【0005】そこで、例えば特開平6-40074号公

報に開示されている技術のように、一旦PDLをプリンタの解像度より高い解像度の2値ラスター画像に変換し、該ラスター画像を解像度変換によってプリンタの解像度と同じ解像度の多値ラスター画像に変換して、画素値に応じてドットの大きさを変える方法が提案されている。しかしながら、後者の技術では、一旦PDLをプリンタの解像度より高い解像度の2値ラスター画像に変換する必要があるため、前者の技術と比較すると多大な記憶領域を要する。そこで、特開平6-168334号公報に開示されている技術のように、一旦PDLをプリンタの解像度より高い解像度の2値ラスター画像を表す2値コードデータに変換し、2値コードデータをプリンタの解像度と同じ解像度の多値ラスター画像を表す多値コードデータに変換して、多値コードデータを多値ラスター画像に変換する方法が提案されている。

【0006】

【発明が解決しようとする課題】ところで、プリンタでプリントアウトするドキュメントには、文字や線画以外にもグラフィクスや写真等の絵柄が混在することがある。高い画質のプリントアウトを得るためには、文字や線画には解像性が要求され、絵柄には階調性が要求される。したがって、文字や線画には上述した技術のようにジャギーを低減する処理が適用され、絵柄には面積階調処理が適用される。

【0007】また、さらに画質を高めるためには、例えばグラフィクスと写真とで面積階調処理の内容を異ならせるといったように、分類をさらに細かくし、その分類毎に適用する処理を変更する手法もある。しかしながら、ラスター画像から文字や線画、あるいはグラフィクスや写真等を完全に分離することは困難である。したがって、従来技術では、文字や線画に面積階調処理が適用されたり、絵柄にジャギーを低減する処理が適用される場合があり、画質向上を図ることが難しいという問題があった。

【0008】この発明は上述した事情に鑑みてなされたもので、ページ記述言語で記述された画像情報をラスター画像に変換する際に必要な記憶容量を少なくでき、より高画質のプリントアウトを得ることができる画像処理装置を提供することを目的としている。

【0009】

【課題を解決するための手段】上述した問題点を解決するために、請求項1記載の発明では、ページ記述言語で記述された画像情報を入力する入力手段と、前記入力手段で入力された前記画像情報の中から文字または線画部を抽出する抽出手段と、前記抽出手段で抽出した文字または線画部を、ラスター展開するための中間的なコードであり、2値の階調を有する第1の中間コードに変換する第1の変換手段と、前記第1の変換手段により変換された第1の中間コードを、当該第1の中間コードより低い解像度で多値の階調を有する第2の中間コードに変換

する第2の変換手段と、前記第1の変換手段により変換された第1の中間コードまたは前記第2の変換手段により変換された第2の中間コードに基づいて、前記入力手段で入力された前記画像情報の所定の領域毎に、当該領域を占める画像の種類を表すタグ情報を生成するタグ生成手段とを具備し、前記タグ生成手段が生成する前記タグ情報は、前記第2の中間コードで表された文字または線画部であることを表す情報を少なくとも含むことを特徴とする。

10 【0010】この発明によれば、ページ記述言語で表され、入力手段から入力された画像情報の中から抽出手段により文字または線画部を抽出し、抽出した文字または線画部を第1の変換手段により2値の階調を有する第1の中間コードに変換し、第1の中間コードを第2の変換手段により上記第1の中間コードより低い解像度で多値の階調を有する第2の中間コードに変換する。また、タグ生成手段により、第1の中間コードまたは第2の中間コードに基づいて、入力された画像情報の所定の領域ごとに当該領域を占める画像の種類を表し、当該第2の中間コードで表された文字または線画部であることを表す情報を少なくとも含むタグ情報を生成する。したがって、ページ記述言語で記述された画像情報をラスター画像に変換する際に必要な記憶容量を少なくすることが可能となるとともに、より高画質のプリントアウトを得ることが可能となる。

【0011】

【発明の実施の形態】次に、図面を参照してこの発明の実施形態について説明する。なお、以下において、画像データは各色成分の1画素あたりの階調数は「256」であり、末尾に「b」が付加されている数は2進表記、「h」が付加されている数は16進表記、何も付加されていない数は10進表記とする。また、以下においては、1つの色成分の処理のみ説明するが、複数色成分の処理については、1つの色成分の処理を色成分の数だけ適用するか、あるいは全ての色成分を同時平行に処理してもよい。

【0012】まず、画像データをランレングスコーディングした形式のコードデータについて説明する。コードデータは、ラスター画像をライン単位でランレングスコーディングしたものであり、ある値の画素が主走査方向のある位置からどれだけ連続するかを記述したものである。例えば、画素値「255（黒画素）」の画素がライン先頭から10番目の画素から20画素連続している場合、「255」、「10」、「20」の3つの整数値が1組のデータとして記述される。このとき、画素値「0」の画素については記述する必要がないため、ラスター画像に比較するとデータ量を圧縮することができる。また、2値画像を対象とした場合には、画素値についても記述する必要がないため、さらにデータ量を圧縮することができる。一方、ラスター画像での種々の画像

処理は、コードデータ間の操作に置き換えられるので、より高速に処理を行うことができる。以下、上述したコードデータをエッジリストと呼ぶこととし、エッジリストの具体例およびエッジリスト間の操作については後述する。

【0013】A. 実施形態の構成

図1は、本発明の実施形態による画像処理装置の一構成例を示すブロック図である。図において、11はコマンド分離部、12は多値文字処理部、13は2値文字処理部、14は多値ベクトル処理部、15は2値ベクトル処理部、16はグラフィクス処理部、17はラスターイメージ処理部、18は合成部である。コマンド分離部11は、入力端Aに入力されたPDLを、多値文字描画コマンド、2値文字描画コマンド、多値ベクトル描画コマンド、2値ベクトル描画コマンド、グラフィクス描画コマンド、ラスターイメージ描画コマンドの6種類のコマンドに分類し、各々、多値文字処理部12、2値文字処理部13、多値ベクトル処理部14、2値ベクトル処理部15、グラフィクス処理部16、ラスターイメージ描画コマンド17に供給する。

【0014】なお、多値文字描画コマンドおよび2値文字描画コマンドは、ASCIIコード等の文字コードと、書体名・ポイント数・フェース・ウェイト・色等の印字特性を示す情報とからなり、さらに、文字描画を多値または2値のいずれで行うかを示す情報も含む。また、多値ベクトル描画コマンドおよび2値ベクトル描画コマンドは、直線・曲線等の線を表す情報であり、始点・終点・経由点等の座標情報および線幅・線の種類・線の色等の画情報とからなり、さらに、ベクトル描画を多値または2値のいずれで行うかを示す情報も含む。また、グラフィクス描画コマンドは、領域を塗り潰す色を表す情報であり、塗り潰す領域の座標情報および塗り潰す色・塗り潰す方法を表す情報を含む。また、ラスターイメージ描画コマンドは、ラスター画像をPDLで表現したものであり、主走査および副走査方向の画素数・解像度・色成分構成等の情報を含む。

【0015】多値文字処理部12は、入力された多値文字描画コマンドを、プリンタ解像度の n 倍（ n は2以上の整数）の解像度の2値のエッジリスト（以下、高解像度エッジリストという）に変換した後、該エッジリストをプリンタ解像度と同じ解像度の多値のエッジリスト（以下、多値エッジリストという）に変換して合成部18に供給する。2値文字処理部13は、入力された2値文字描画コマンドを、プリンタ解像度と同じ解像度の2値のエッジリスト（以下、2値エッジリストという）に変換し、合成部18に供給する。多値ベクトル処理部14は、入力された多値ベクトルコマンドを、高解像度エッジリストに変換した後、該エッジリストを多値エッジリストに変換して合成部18に供給する。

【0016】2値ベクトル処理部15は、入力されたベ

クトルコマンドを、2値エッジリストに変換して合成部18に供給する。グラフィクス処理部16は、入力されたグラフィクス描画コマンドを、多値エッジリストに変換して合成部18に供給する。ラスターイメージ処理部17は、入力されたラスターイメージ描画コマンドを、多値エッジリストに変換して合成部18に供給する。合成部18は、多値文字処理部12、2値文字処理部13、多値ベクトル処理部14、2値ベクトル処理部15、グラフィクス処理部16、ラスターイメージ処理部17から供給されるエッジリストを合成するとともに、ラスター画像に変換して出力端Aに出力する。また、ラスター画像への変換と同時にラスター画像の各画素の分類を示す分類データを生成し、出力端Bに出力する。なお、分類データには、エッジリストがどの処理部から出力されたかを示す情報とともに、異なる処理部から出力されたエッジリストがそれぞれ表すラスター画像の境界を示す情報も含まれる。

【0017】ここで、図2は、PDLの一例を示す概念図である。以下、図2に例示するPDLが入力端Aに入力された場合を例にとってさらに詳細に説明する。なお、図2において、TYPEは文字描画コマンド、LINEは直線描画コマンド、PAINTはグラフィクス描画コマンド、RASTERはラスターイメージ描画コマンドである。また、コマンド名に続いて括弧（）で括られた部分は各コマンドのパラメータ、括弧{}で括られた部分はラスターイメージデータである。TYPEコマンドの4つのパラメータは、先頭から順に描画開始 x 座標、描画開始 y 座標、フォントの種類、ポイント数、描画文字列、描画モードであり、図2の1行目は、文字列「a b c d e f g」を座標（ x_1, y_1 ）を始点として多値モードで、2行目は、2値モードで描画することを意味する。

【0018】また、LINEコマンドの6つのパラメータは、先頭から順に描画開始 x 座標、描画開始 y 座標、描画終了 x 座標、描画終了 y 座標、線幅、描画モードであり、図2の3行目は、座標（ x_2, y_2 ）から（ x_3, y_3 ）に1ポイントの太さの直線を多値モードで、4行目は、2値モードで描画することを意味する。次に、図2の5行目は、座標（ x_4, y_4 ）と座標（ x_5, y_5 ）とを結ぶ直線を対角線とする矩形領域を、色空間をRGBとして指定色RED（赤色）で塗り潰すことを意味する。また、図2の6行目は、座標（ x_6, y_6 ）を左上隅として、 p 画素 \times 1ラインのラスターイメージを、色空間をRGBとし、主走査方向の解像度 r_x 、副走査方向の解像度 r_y で括弧{}内のラスターイメージデータにしたがって描画することを意味する。なお、上記コマンドおよび各コマンドのパラメータは一例であり、コマンド名、各コマンドのパラメータ内容は、上記内容に限定されない。また、各描画動作に対するコマンドの設定も上記内容に限らず、例えばTYPEコマ

ンドを2つ以上のコマンドに分割するといったように変更してもよい。

【0019】コマンド分離部11は、多値文字描画コマンドとして図2に示す1行目を多値文字処理部12へ、2値文字描画コマンドとして2行目を2値文字処理部13へ、多値ベクトル描画コマンドとして3行目を多値ベクトル処理部14へ、2値ベクトル描画コマンドとして4行目を2値ベクトル処理部15へ、グラフィクス描画コマンドとして5行目をグラフィクス処理部16へ、ラ

スターイメージ描画コマンドとして6行目をラスターイメージ処理部17へそれぞれ出力する。

【0020】多値文字処理部12は、多値文字描画コマンドとして入力された1行目のコマンドを解釈し実行する。すなわち、文字列を指定フォントで多値モードで描画する。フォントには、文字をラスター画像データで記述したビットマップフォントと、文字の輪郭を直線・曲線等の線の集合体として線の描画情報（アウトラインデータ）で記述したアウトラインフォントとがあるが、ポイント数・フェース・ウェイトに対応した変形処理の容易さと変形処理後のディフェクトの少なさとから、本実施形態ではアウトラインフォントを使用することとする。アウトラインフォントを使用した場合、指定された文字のアウトラインデータが読み出され、該アウトラインデータに基づいて仮想座標平面に文字の輪郭が描画され、該輪郭で囲まれた閉領域が塗り潰されて仮想座標平面上で理想的な文字画像を得る。そして、上記仮想座標平面上の文字の輪郭（以下、単に文字輪郭という）に基づいて多値エッジリストを生成し、合成部18に出力する。

【0021】2値文字処理部13は、2値文字描画コマンドとして入力された2行目のコマンドを解釈し実行する。すなわち、文字列を指定フォントで2値モードで描画する。ここでは、多値文字処理部12と同様にして仮想座標平面上で理想的な文字画像を得る。そして、文字輪郭に基づいて2値エッジリストを生成し、合成部18に出力する。多値ベクトル処理部14は、多値ベクトル描画コマンドとして入力された3行目のコマンドを解釈し実行する。すなわち、ベクトルを多値モードで描画する。まず、仮想座標平面上で指定された2つの座標点を端点とする直線が描画され、該直線から指定された線幅の半分の距離にある領域が塗り潰されて仮想座標平面上で理想的なベクトル画像を得る。そして、上記仮想座標平面上のベクトルの輪郭（以下、単にベクトル輪郭）に基づいて多値エッジリストを生成し、合成部18に出力する。

【0022】次に、2値ベクトル処理部15は、2値ベクトル描画コマンドとして入力された4行目のコマンドを解釈し実行する。すなわち、ベクトルを2値モードで描画する。ここでは、多値ベクトル処理部14と同様にして仮想座標平面上で理想的なベクトル画像を得る。そ

して、上記仮想座標平面上のベクトルの輪郭に基づいて2値エッジリストを生成し、合成部18に出力する。グラフィクス処理部16は、グラフィクス描画コマンドとして入力された5行目のコマンドを解釈し実行する。すなわち、指定された領域を指定された色で塗り潰す。まず、仮想座標平面上で塗り潰し領域が設定され、該領域を指定された色で塗り潰す。そして、上記仮想座標平面上の塗り潰された領域に基づいて2値エッジリストを生成し、合成部18に出力する。なお、塗り潰しの際に指定された色空間をプリンタに適合した色空間に変換するのは当然であり、さらにその際にプリンタの色再現特性に応じて色の近似処理を行う。この色空間変換処理および色近似処理の内容はどのようなものでも構わない。

【0023】次に、ラスターイメージ処理部17は、ラスターイメージ描画コマンドとして入力された6行目のコマンドを解釈し実行する。すなわち、指定されたラスターイメージデータに従って指定された領域に指定された色空間および指定された解像度で描画する。まず、仮想座標平面上に主走査および副走査方向の画素数および解像度に応じた描画領域が設定され、該領域に指定された色空間で指定されたラスターイメージデータに従ってラスターイメージが描画される。なお、塗り潰しの際に指定された色空間をプリンタに適合した色空間に変換するのは当然であり、さらにその際にプリンタの色再現特性に応じて色の近似処理を行う。また、指定された解像度とプリンタの解像度とが一致しない場合には、解像度変換処理を行う。この色空間変換処理、色近似処理および解像度変換処理の内容はどのようなものでも構わない。

【0024】合成部18は、多値文字処理部12、2値文字処理部13、多値ベクトル処理部14、2値ベクトル処理部15、グラフィクス処理部16およびラスターイメージ処理部17から供給されるエッジリストを1つのエッジリストに合成し、ラスター画像に変換して出力端Aから外部に出力する。また、合成部18に入力される各エッジリストがどの処理部から出力されたかを示すデータを画素毎に作成して上記ラスター画像と同期を取りながら出力端Bから外部に出力する。合成部18での合成処理は、エッジリスト間の操作によって実現される。

【0025】B. 実施形態の動作

次に、上述した実施形態の動作について説明する。まず、多値文字処理部12での仮想座標平面上の文字の輪郭からの多値エッジリスト生成について詳述する。図3は、エッジリスト生成過程を説明するためのフローチャートである。まず、描画する文字に対応したアウトラインデータをロードし（ステップS100）、高解像度エッジリストを作成し（ステップS101）、該エッジリストを多値エッジリストに変換する（ステップS102）。TYPEコマンドで指定された文字列中の全ての

文字の描画が終了すると、合成部18に多値エッジリストを出力して処理を終え、描画していない文字がある場合には、ステップS100以降を繰り返す。なお、以下においては $n=2$ として説明する。

【0026】図4は、図3におけるステップS101での処理の詳細を示すフローチャートである。なお、図4において、 i は主走査方向の処理位置、 j は副走査方向の処理位置を示すインデックスであり、 st は主走査方向の塗り潰し開始位置（以下、塗り潰し開始位置という）、 $r1$ は塗り潰す画素の連続数（以下、塗り潰し画素連続数という）を格納する変数であり、 $f1$ はその値が「1」のときは塗り潰し開始位置 st および連続数 $r1$ に有効データが残っており「0」のときは残っていないことを示すフラグである。また、 P および L は、1つの文字を描画する領域の主走査方向および副走査方向の画素数である。なお、記号 $=$ は右辺の数または演算結果を左辺の変数に格納することを、記号 $==$ は両辺が等しいことを意味する。

【0027】まず、処理に必要な変数を初期化する（ステップS200、S201）。次に、インデックス i および j で示される仮想座標点 $TP(i, j)$ が塗り潰し領域にあるかどうかを判定し、塗り潰し領域にあるときは、ステップS203以降の処理を行い、塗り潰し領域にないときは、ステップS206以降の処理を行う（ステップS202）。なお、仮想座標点 $TP(i, j)$ は、プリンタの解像度の2倍の解像度の画素の中心点である。仮想座標点 $TP(i, j)$ が塗り潰し領域にあるときは、塗り潰し画素連続数 $r1$ をインクリメントし（ステップS203）、フラグ $f1$ が「0」か否か、つまり塗り潰し開始位置 st および塗り潰し画素連続数 $r1$ に有効データが残っていないかどうかを調べ（ステップS204）、残っていなければ、現在の主走査方向の位置を主走査方向の塗り潰し開始位置 st に設定する（ステップS205）。

【0028】一方、仮想座標点 $TP(i, j)$ が塗り潰し領域にないときは、フラグ $f1$ が「1」か否か、つまり塗り潰し開始位置 st および塗り潰し画素連続数 $r1$ に有効データが残っているかどうかを調べ（ステップS206）、残っていれば塗り潰し開始位置 st および塗り潰し画素連続数 $r1$ をリストに登録し（ステップS207）、塗り潰し開始位置 st 、塗り潰し画素連続数 $r1$ およびフラグ $f1$ を初期化する（ステップS208）。ステップS205またはステップS208が終了するか、ステップS204またはステップS206での判定結果が「NO」である場合には、インデックス i をインクリメントし（ステップS209）、主走査方向画素数 P を2倍した値とインデックス i とを比較して（ステップS210）、等しいならば、ステップS211以降の処理を行い、等しくないならば、ステップS202以降の処理を行う。

【0029】インデックス i が主走査方向画素数 P を2

倍した値に等しい場合には、フラグ $f1$ が「1」か否か、つまり塗り潰し開始位置 st および塗り潰し画素連続数 $r1$ に有効データが残っているかどうかを調べ（ステップS211）、残っていれば塗り潰し開始位置 st および塗り潰し画素連続数 $r1$ をリストに登録する（ステップS212）。そして、ステップS212での処理が終了するか、ステップS211での判定結果が「NO」である場合には、リストに終端符号 EOL を登録してインデックス j をインクリメントし（ステップS213）、副走査方向画素数 L を2倍した値とインデックス j とを比較して（ステップS214）、等しいならば処理を終了し、等しくないならばステップS201以降の処理を行う。上述した処理により、プリンタ解像度の2倍の解像度の2値のエッジリストが作成される。なお、上述した処理においては、作成するエッジリストは2値であるため、印字する画素の開始位置および連続数のみがリストの内容となる。また、ライン終了時に終端符号 EOL をリストに登録していることから分かるように、エッジリストはライン単位で作成される。

【0030】次に、図5ないし図7は、図3のステップS102での処理の詳細を示すフローチャートである。なお、図5ないし図7において、 i は主走査方向の処理位置、 j は副走査方向の処理位置を示すインデックスであり、 $st1 \sim st6$ は主走査方向の塗り潰し開始位置、 $r11 \sim r16$ は塗り潰す画素の連続数を格納する変数であり、 st は多値エッジリストに登録する主走査方向の塗り潰し開始位置、 $r1$ は多値エッジリストに登録する塗り潰す画素の連続数、 pv は多値エッジリストに登録する画素値を格納する変数である。また、 $n2$ は注目画素、 $n1$ および $n3$ はその直上・直下の画素の状態を表す変数であり、 m は画素値を格納する変数である。また、 P および L が表す数、記号 $=$ および記号 $==$ の意味は図4と同様であり、関数 $LUT-A$ および関数 $LUT-B$ は互いに異なるルックアップテーブル（以下、 LUT という）を用いて引数を変換する関数を意味する。

【0031】まず、インデックス i に初期値を設定し（ステップS300）、先頭ラインの処理を行う（ステップS301）。なお、先頭ラインの処理については後述する。次に、インデックス i をインクリメントし（ステップS302）、 $ST1, RL1 \sim ST6, RL6$ を読み込むとともに、処理に必要な変数を初期化する（ステップS303）。なお、 STx, RLx （ $x=1 \sim 6$ ）は、図3のステップS102で作成した2値のエッジリストの（ $2 \cdot j + x - 3$ ）番目のラインのリストから読み込んだ主走査方向の塗り潰し開始位置および塗り潰す画素の連続数である。例えば、 $j=1$ のとき $ST1, RL1$ は0番目のライン（つまり先頭ライン）のリストから読み込んだ主走査方向の塗り潰し開始位置および塗り潰す画素の連続数である。次に、画素の状態を表

す変数n1、n3を算出する(ステップS304)。表 *【0032】
1に算出した値と画素の状態を示す。 * 【表1】

| 値 | 状態 | 値 | 状態 | 値 | 状態 | 値 | 状態 |
|---|----|---|----|----|----|----|----|
| 0 | | 4 | | 8 | | 12 | |
| 1 | | 5 | | 9 | | 13 | |
| 2 | | 6 | | 10 | | 14 | |
| 3 | | 7 | | 11 | | 15 | |

表1において、状態欄の図形は、図3のステップS101で作成した2値のエッジリストのうち、注目画素に対応する部分の状態を模式的に示したものであり、白色矩形は塗り潰さない画素、黒色画素は塗り潰す画素を表す。なお、この算出方法については後述する。

【0033】次に、変数n2が「3」に等しいかどうかを調べ(ステップS305)、等しければステップS306以降の処理を行い、等しくなければステップS309※

※以降の処理を行う。変数n2が「3」に等しい場合、変数n3が「0」または「12」に等しいかどうかを調べ(ステップS306)、どちらかに等しい場合には変数n2を表3に示すLUTで濃度値に変換して変数mに格納する(ステップS307)。一方、どちらにも等しくない場合には変数n2を表2に示すLUTで濃度値に変換して変数mに格納する(ステップS308)。

【表2】

| 入力 | 出力 | 値 | 状態 | 値 | 状態 | 値 | 状態 |
|----|-----|---|-----|----|-----|----|-----|
| 0 | 0 | 4 | 64 | 8 | 64 | 12 | 128 |
| 1 | 64 | 5 | 128 | 9 | 128 | 13 | 192 |
| 2 | 64 | 6 | 128 | 10 | 128 | 14 | 192 |
| 3 | 128 | 7 | 192 | 11 | 192 | 15 | 255 |

【0034】

★30★【表3】

| 入力 | 出力 | 値 | 状態 | 値 | 状態 | 値 | 状態 |
|----|-----|---|-----|----|-----|----|-----|
| 0 | 0 | 4 | 64 | 8 | 64 | 12 | 192 |
| 1 | 64 | 5 | 128 | 9 | 128 | 13 | 192 |
| 2 | 64 | 6 | 128 | 10 | 128 | 14 | 192 |
| 3 | 192 | 7 | 192 | 11 | 192 | 15 | 255 |

【0035】一方、変数n2が「3」に等しくない場合、変数n2が「12」に等しいかどうかを調べ(ステップS309)、等しければステップS310以降の処理を行い、等しくなければステップS312以降の処理を行う。変数n2が「12」に等しい場合、変数n1が「0」または「3」に等しいかどうかを調べ(ステップS310)、等しければ表3に示すLUTで濃度値に変換して変数mに格納し(ステップS311)、等しくなければ表2に示すLUTで濃度値に変換して変数mに格納する(ステップS312)。

【0036】上記ステップS307、ステップS308、ステップS311、ステップS312のいずれかで☆50

☆の処理が終了すると、濃度値が格納されている変数mと変数pvが「0」より大きいかどうかを調べる(ステップS310)。そして、変数mと変数pvがともに「0」より大きければ、変数mと変数pvが等しいかどうかを調べ(ステップS312)、等しければ変数r1をインクリメントし(ステップS313)、等しくなければ変数st、変数pvおよび変数r1をリストに登録し(ステップS314)、変数stにインデックスi、変数pvに変数m、変数r1に「1」を設定する(ステップS315)。

【0037】一方、変数mが「0」であり、変数pvが「0」より大きければ、変数st、変数pvおよび変数

rlをリストに登録し(ステップS317)、変数pvに「0」を設定する(ステップS318)。また、変数mが「0」より大きく、変数pvが「0」であれば、変数stにインデックスi、変数pvに変数m、変数rlに「1」を設定する(ステップS316)。つまり、ステップS310からステップS319における処理は、ランレングスコーディング処理である。

【0038】次に、ステップS313、ステップS315、ステップS316またはステップS318での処理が終了するか、ステップS310で変数mと変数pvがともに「0」であった場合には、インデックスiをインクリメントし(ステップS319)、インデックスiがPに等しいかどうかを調べる(ステップS320)。そして、等しければステップS321以降の処理を行い、等しくなければステップS304以降の処理を行う。

【0039】インデックスiがPに等しい場合、変数pvが「0」より大きいかどうかを調べ(ステップS321)、大きければ変数st、変数pvおよび変数rlをリストに登録する(ステップS322)。そして、ステップS322での処理が終了するか、ステップS321での判定結果が「NO」であった場合、EOLをリストに登録してインデックスjをインクリメントし(ステップS323)、インデックスjがLから「1」を減じた値に等しいかどうかを調べる(ステップS324)。そして、等しければ後述する最終ライン処理を行って(ステップS325)、全ての処理を終了し、等しくなければステップS303以降の処理を行う。

【0040】ここで、図5のステップS301における先頭ライン処理について説明する。図8および図9は、先頭ライン処理の詳細を示すフローチャートである。なお、図6において、図5ないし図7と同じ名前のインデックス、変数、記号および関数は、図5ないし図7と同様である。

【0041】まず、ST3、RL3～ST6、RL6を読み込むとともに、処理に必要な変数を初期化する(ステップS400)。次に、画素の状態を表す変数n2、n3を算出する(ステップS401)。次に、変数n2が「3」に等しいかどうかを調べ(ステップS402)、等しければステップS403以降の処理を行い、等しくなければステップS405以降の処理を行う。変数n2が「3」に等しい場合には、変数n3が「0」または「12」に等しいかどうかを調べ(ステップS403)、どちらかに等しい場合には、変数n2を表3に示すLUTで濃度値に変換して変数mに格納する(ステップS404)。一方、どちらにも等しくない場合には、変数n2を表2に示すLUTで濃度値に変換して変数mに格納する(ステップS405)。

【0042】上記ステップS404、ステップS405のいずれかでの処理が終了すると、濃度値が格納されている変数mと変数pvが「0」より大きいかどうかを調

べ(ステップS406)、変数mと変数pvとがともに「0」より大きければ、変数mと変数pvとが等しいかどうかを調べる(ステップS407)。そして、等しければ、変数rlをインクリメントし(ステップS408)、等しくなければ、変数st、変数pvおよび変数rlをリストに登録し(ステップS409)、変数stにインデックスi、変数pvに変数m、変数rlに「1」を設定する(ステップS410)。一方、変数mが「0」であり、変数pvが「0」より大きければ、変数st、変数pvおよび変数rlをリストに登録し(ステップS412)、変数pvに「0」を設定する(ステップS413)。また、変数mが「0」より大きく、変数pvが「0」であれば、変数stにインデックスi、変数pvに変数m、変数rlに「1」を設定する(ステップS411)。つまり、ステップS406からステップS413における処理は、先頭ラインのランレングスコーディング処理である。

【0043】次に、ステップS408、ステップS410、ステップS411またはステップS413での処理が終了するか、ステップS406で変数nと変数pvがともに「0」であった場合には、インデックスiをインクリメントし(ステップS414)、インデックスiがPに等しいかどうかを調べ(ステップS415)、等しければステップS416以降の処理を行い、等しくなければステップS401以降の処理を行う。インデックスiがPに等しい場合、変数pvが「0」より大きいかどうかを調べ(ステップS416)、大きければ変数st、変数pvおよび変数rlをリストに登録する(ステップS417)。そして、ステップS417での処理が終了するか、子ステップS416での判定結果が「NO」であった場合、EOLをリストに登録して(ステップS418)、処理を終了する。

【0044】次に、図7のステップS325における最終ライン処理について説明する。図10および図11は、最終ライン処理の詳細を説明するためのフローチャートである。なお、図10および図11において、図5ないし図7と同じ名前のインデックス、変数、記号および関数は、図5ないし図7と同様である。

【0045】まず、ST1、RL1～ST4、RL4を読み込むとともに、処理に必要な変数を初期化する(ステップS500)。次に、画素の状態を表す変数n1～n2を算出する(ステップS501)。次に、変数n1が「12」に等しいかどうかを調べ(ステップS502)、等しければステップS503以降の処理を行い、等しくなければステップS505以降の処理を行う。変数n2が「12」に等しい場合には、変数n1が「0」または「3」に等しいかどうかを調べ(ステップS503)、どちらかに等しい場合には、変数n2を表3に示すLUTで濃度値に変換して変数mに格納する(ステップS504)。一方、どちらにも等しくない場合には、

変数n2を表2に示すLUTで濃度値に変換して変数mに格納する(ステップS505)。

【0046】上記ステップS504、ステップS505のいずれかでの処理が終了すると、濃度値が格納されている変数mと変数pvが「0」より大きいかどうかを調べ(ステップS506)、変数mと変数pvがともに「0」より大きければ変数mと変数pvとが等しいかどうかを調べ(ステップS507)、等しければ変数r1をインクリメントする(ステップS508)。一方、等しくなければ変数st、変数pvおよび変数r1をリストに登録し(ステップS509)、変数stにインデックスi、変数pvに変数m、変数r1に「1」を設定する(ステップS510)。一方、変数mが「0」であり、変数pvが「0」より大きければ、変数st、変数pvおよび変数r1をリストに登録し(ステップS513)、変数pvに「0」を設定する(ステップS514)。また、変数mが「0」より大きく、変数pvが「0」であれば、変数stにインデックスi、変数pvに変数m、変数r1に「1」を設定する(ステップS512)。つまり、ステップS506からステップS514における処理は、最終ラインのランレングスコーディング処理である。

【0047】次に、ステップS508、ステップS510、ステップS512またはステップS514での処理が終了するか、ステップS506で変数mと変数pvとがともに「0」であった場合には、インデックスiをインクリメントし(ステップS514)、インデックスiがPに等しいかどうかを調べる(ステップS515)。そして、双方が等しければステップS517以降の処理を、等しくなければステップS501以降の処理を行う。インデックスiがPに等しい場合、変数pvが「0」より大きいかどうかを調べ(ステップS517)、大きければ変数st、変数pvおよび変数r1をリストに登録する(ステップS518)。ステップS518での処理が終了するか、ステップS517での判定結果が「NO」であった場合には、EOLをリストに登録して(ステップS519)、処理を終了する。

【0048】次に、図5のステップS304、図8のステップS401および図10のステップS501における変数n1、n3の算出方法について説明する。なお、この算出方法は、変数n1～n3について同じであるため、変数n1の場合についてのみ説明する。図12および図13は、変数n1を算出するアルゴリズムを説明するためのフローチャートである。なお、図12および図18において、図ないし図7と同じ名前のインデックス、変数および記号は、図5ないし図7と同様であり、記号「|」はビットOR演算を示すものとする。

【0049】まず、変数n1を初期化し(ステップS600)、注目画素の主走査方向での位置を示すインデックスiが変数ST1以上、かつ変数ST1と変数RL1

との和より小さいか否かを調べる(ステップS601)。つまり、注目画素が塗潰領域にあるかどうかを調べる。ステップS601での判定結果が「YES」の場合、すなわち注目画素が塗潰領域にある場合には、変数n1と定数「1000b」とのビットOR演算結果を変数n1に格納する(ステップS602)。一方、「NO」の場合、すなわち注目画素が塗潰領域にない場合には、インデックスiが変数ST1と変数RL1の和以上であるか否かを調べる(ステップS603)。ステップS603での判定結果が「YES」の場合には、リストからST1およびRL1を読み込む(ステップS604)。

【0050】次に、ステップS602またはステップS604での処理が終了するか、ステップS603での判定結果が「NO」である場合には、インデックスiに「1」を加えた値が変数ST1以上、かつ変数ST1と変数RL1との和より小さいか否かを調べる(ステップS605)。つまり、注目画素の右の画素(以下、単に右画素という)が塗潰領域にあるか否かを調べる。ステップS605での判定結果が「YES」の場合、すなわち右画素が塗潰領域にある場合には、変数n1と定数「100b」とのビットOR演算結果を変数n1に格納する(ステップS606)。一方、「NO」の場合、すなわち右画素が塗潰領域にない場合には、インデックスiに「1」を加えた値が変数ST1と変数RL1との和以上であるか否かを調べる(ステップS607)。そして、ステップS607での判定結果が「YES」の場合には、リストからST1およびRL1を読み込む(ステップS608)。

【0051】次に、ステップS606またはステップS608での処理が終了するか、ステップS607での判定結果が「NO」である場合には、インデックスiが変数ST2以上で、かつ変数ST2と変数RL2との和より小さいか否かを調べる(ステップS609)。つまり、注目画素の下画素(以下、単に下画素という)が塗潰領域にあるかどうかを調べる。ステップS609での判定結果が「YES」の場合、すなわち下画素が塗潰領域にある場合には、変数n1と定数「10b」とのビットOR演算結果を変数n1に格納する(ステップS610)。一方、「NO」の場合、すなわち下画素が塗潰領域にない場合には、インデックスiが変数ST2と変数RL2との和以上であるか否かを調べる(ステップS611)。そして、ステップS611での判定結果が「YES」の場合には、リストからST2およびRL2を読み込む(ステップS612)。

【0052】次に、ステップS610またはステップS612での処理が終了するか、ステップS611での判定結果が「NO」である場合には、インデックスiに「1」を加えた値が変数ST2以上で、かつ変数ST2と変数RL2との和より小さいか否かを調べる(ステッ

ブS613)。つまり、注目画素の右下の画素（以下、単に右下画素という）が塗潰領域にあるかどうかを調べる。ステップS613での判定結果が「YES」の場合、すなわち右下画素が塗潰領域にある場合には、変数n1と定数「1b」とのビットOR演算結果を変数n1に格納し（ステップS614）、当該処理を終了する。一方、「NO」の場合、すなわち次画素が塗潰領域にない場合には、インデックスiに「1」を加えた値を変数ST2と変数RL2との和以上であるか否かを調べる（ステップS615）。ステップS615での判定結果が「YES」の場合には、リストからST2およびRL2を読み込み（ステップS616）、当該処理を終了する。また、ステップS615での判定結果が「NO」である場合には、何もせずに当該処理を終了する。なお、上記処理において、画素の相対的な位置関係は、ラインの先頭画素を最左端、先頭ラインを最上端ラインとして記述している。

【0053】ここで、明朝体の文字「一」を例にとり、処理の具体例を示す。図14は、処理過程でのラスターライズ画像ならびに比較対照のための概念図である。また、図15は、図14(a)、(e)の部分拡大図である。なお、図14および図15において、矩形は画素を表し、白色矩形以外の矩形の意味についてはそれぞれ右横の注釈の通りである。また、図14においては、1つの文字を主走査方向16画素、副走査方向16画素の領域に展開するものとし、該領域の左上端を座標原点とし、上述した画素の相対的な位置関係と同様の位置関係で示している。また、図15においては、破線矢印は、プリンタ解像度のn倍の解像度の2値の画像データを表すエッジリストを作成する際に仮想的に設定されるスキャンラインであり、主走査および副走査の両方向のスキャンラインの交点が図4における仮想座標点TP(i, j)である。なお、図15では、上述と同様にn=2の場合について示しているが、nの値が他の値であっても同様に仮想座標点を設定できるのは明らかである。

【0054】図14(a)は、文字「一」を理想的に展開した場合の塗潰領域を示す概念図であるが、固有の解像度を有するプリンタに出力するためには、塗潰領域を含む画素の値を何らかの論理に従って決定する必要がある。例えば、図14(a)に示す塗潰領域を少しでも含む画素を印字、含まない画素を非印字と決定すると、図14(b)に示すような印字結果が得られる。両図を見てもわかるように、この画素値決定論理では、高い画質の印字結果を得ることはできない。一方、前述した従来技術に記載されているように、PDLから作成したプリンタ解像度のn倍の解像度の2値の画像データを表すエッジリストから単純にプリンタ解像度の多値の画像データを表すエッジリストに変換してラスターライズすると、図14(c)に示すラスター画像が得られ、このラスター画像を印字すると図14(d)に示す印字結果が得ら

れる。この場合、特に文字「一」の横棒部分が想定していた線幅よりも細い線幅で印字されてしまい、場合によっては途切れたり印字されないことがある。

【0055】一方、上述した本発明の実施形態によってラスターライズすると、図14(e)に示すラスター画像が得られ、このラスター画像を印字すると図14(f)に示す印字結果が得られる。図15(a)および図15(b)は、それぞれ図14(a)および図14(e)の部分拡大図である。以下、本発明によるラスターライズ処理例を図14(a)、図14(e)、図14(f)および図15を参照しながら説明する。まず、本実施形態において、入力されたPDLのうち、多値展開を指定された文字についてPDLからアウトラインデータ、さらにエッジリストへと変換が行われる。図4ないし図7に示したアルゴリズムに従って、各仮想座標点が塗潰領域にあるか否かが判定され、その結果エッジリストが作成されていく。

【0056】例えば、図15(a)の7ライン目について見てみると、上側の主走査方向のスキャンライン（以下、単に主走査スキャンラインという）では、13番目の画素の右側の仮想座標点が塗潰領域内にある。これは、図4のステップS202において、 $i=27$ 、 $j=14$ のときに判定結果が「YES」となることを意味している。この仮想座標点までに、ステップS202における判定結果が「YES」となることはなく、また、この仮想座標点以降の同じ主走査スキャンライン上の仮想座標点についてもステップS202における判定結果が「NO」となるため、この主走査スキャンラインが終了した時点でのリストには、 $st=27$ 、 $rl=1$ のみが登録されている。

【0057】一方、下側の主走査スキャンラインでは、13番目の画素の左右両側および14番目の画素の左側の仮想座標点が連続して塗潰領域内にある。従って、上側と同様に処理を行うと、この主走査スキャンラインが終了した時点でのリストには、 $st=26$ 、 $rl=3$ がリストに追加されている。同様に8ライン目についても処理を行うと、上側の主走査スキャンラインでは、 $st=2$ 、 $rl=26$ がリストに追加され、下側の主走査スキャンラインでは、何もリストに追加されない。上述したライン以外については、いずれの仮想座標点も塗潰領域にはないため、最終的には表4に示すようなリストが得られる。

【表4】

| j | st | rl |
|----|----|----|
| 14 | 27 | 1 |
| 15 | 26 | 3 |
| 16 | 2 | 26 |

【0058】表4に示すリストは、一形態例であってリ

ストを見やすい形で表現したものにすぎず、リストの記憶媒体でのデータ形式は、表4に限定されるわけではなく、どの主走査スキャンラインにどのようなst, r1が登録されているかが分かればどのような形態であっても構わない。なお、表4においては、EOL符号のみが登録されている主走査スキャンラインおよびEOL符号については省略している。

【0059】次に、表4に示すリストに基づいて多値のエッジリストが作成される。この作成の具体例を図15(b)の7ライン目および8ライン目を例にとり説明する。まず、7ライン目では、1番目の画素から12番目の画素までは、図12または図13におけるn1、n2の算出でステップS601、ステップS605、ステップS609およびステップS613での判定結果がいずれも「NO」となり、n3の算出でステップS601およびステップS605での判定結果がいずれも「YES」、ステップS609およびステップS613での判定結果がいずれも「NO」となるので、n1=0、n2=0、n3=12となり、図5のステップS305およびステップS309での判定結果がいずれも「NO」となる。従って、変数n2が表2に示すLUTで変換され、画素値「0」が得られる。ゆえに、多値エッジリストには何も登録されない。同ラインの13番目の画素の*

*時点でn1=0、n2=7、n3=12、14番目の画素の時点でn1=0、n2=2、n3=12となるので、それぞれ表2に示すLUTから画素値「192」および画素値「64」が得られる。

【0060】先頭画素および15番目以降の画素については、いずれもn1=0、n2=0、n3=0となるので、7ライン目の処理が終了した時点で、多値エッジリストには、st=13、pv=192、r1=1およびst=14、pv=64、r1=1が登録されている。次に、8ライン目では、1番目の画素から12番目の画素までは、n1=0、n2=12、n3=0となり、図5のステップS305での判定結果は「NO」、ステップS309およびステップS310での判定結果は「YES」となる。従って、変数n2が表3に示すLUTで変換され、画素値「192」が得られる。また、同ラインの13番目の画素の時点で、n1=7、n2=12、n3=0となり、14番目の画素の時点で、n1=2、n2=12、n3=0となり、ともに画素値「128」が得られる。上述したライン以外については、いずれの画素についても、画素値は「0」となるため、最終的には、例えば表5に示すようなリストが得られる。

【表5】

| j | st | pv | r1 |
|---|----|-----|----|
| 7 | 13 | 192 | 1 |
| | 14 | 64 | 1 |
| 8 | 1 | 192 | 12 |
| | 13 | 128 | 2 |

【0061】表5に示すリストは、一形態例であってリストを見やすい形で表現したものにすぎず、リストの記憶媒体でのデータ形式は表5に限定されるわけではなく、どのラインにどのようなst, pv, r1が登録されているかが分かればどのような形態であっても構わない。なお、表5においては、EOL符号のみが登録されている主走査スキャンラインおよびEOL符号については省略している。以上により、多値エッジリストが得られ、多値文字処理部12から合成部18に出力される。

【0062】次に、2値文字処理部13での処理を説明する。まず、描画する文字に対応したアウトラインデータをロードし、2値エッジリストを作成する。TYPEコマンドで指定された文字列中の全ての文字の描画が終了すると、合成部18に2値エッジリストを出力して処理を終了し、描画していない文字がある場合には、アウトラインデータのロード以降を繰り返す。

【0063】なお、上述した処理のうちアウトラインデータのロードは、多値文字処理部12と同じであり、文字輪郭からの2値エッジリストの作成は、多値文字処理部

※部12における高解像度エッジリストの作成と解像度の設定、つまり仮想座標点の設定が異なるのみで他は同じであるので、詳細な説明は省略する。文字「一」から作成したエッジリストを表6に示す。

【表6】

| j | st | r1 |
|---|----|----|
| 7 | 13 | 1 |
| 8 | 1 | 14 |

【0064】次に、多値ベクトル処理部14での処理を説明する。図16は、多値ベクトル処理部14での処理を示すフローチャートである。まず、指定されたベクトルを描画し（ステップS700）、高解像度エッジリストを作成し（ステップS701）、該エッジリストを多値エッジリストに変換する（ステップS702）。全てのベクトルの描画が終了すると、合成部18に多値エッジリストを出力して処理を終了し、描画していないベク

トルがある場合には、ステップS700以降を繰り返す。ステップS701以降は、多値文字処理部12と同じであるので、ここでは、ステップS700についてのみ説明する。まず、指定されたベクトルは、指定された座標点情報に従って仮想座標平面上に描画される。次に、仮想座標平面上に描画されたベクトルを中心として、指定された線幅の半分の距離にある領域を塗潰領域とする。そして、多値文字処理部12と同様に仮想座標点を設定し、高解像度エッジリストの作成、および該エッジリストの多値エッジリストへの変換を行っていく。【0065】ここで、図17は、多値ベクトル処理部14での処理の具体例を説明するための概念図である。まず、指定されたベクトルが図17(a)に示すハッチング領域の中心線のように描画される。次に、該中心線から指定された線幅の半分の距離にある領域が図17(a)に示すハッチング領域のように塗潰領域に指定される。このとき、多値文字処理部12においては、アウトラインデータをロードし、塗潰領域を設定したときと*

*同じ状態であるので、上記処理以降では、多値文字処理部12と同様の処理を行い、一旦、高解像度エッジリストを取得した後、多値エッジリストを取得する。表7は、高解像度エッジリストを示し、表8は、多値エッジリストを示す。また、図17(c)は、最終的に得られた多値エッジリストを印字した結果を示す図である。

【表7】

| j | st | rl |
|---|----|----|
| 1 | 1 | 9 |
| 2 | 1 | 17 |
| 3 | 4 | 20 |
| 4 | 12 | 17 |
| 5 | 17 | 11 |
| 6 | 25 | 3 |

【0066】

【表8】

| j | st | p v | rl |
|---|----|-----|----|
| 1 | 1 | 255 | 4 |
| | 5 | 192 | 1 |
| | 6 | 128 | 3 |
| | 9 | 64 | 1 |
| 2 | 2 | 64 | 1 |
| | 3 | 128 | 3 |
| | 6 | 192 | 1 |
| | 7 | 255 | 5 |
| | 12 | 192 | 1 |
| | 13 | 128 | 2 |
| 3 | 6 | 64 | 1 |
| | 7 | 128 | 3 |
| | 6 | 192 | 1 |
| | 7 | 255 | 1 |

【0067】次に、2値ベクトル処理部15での処理を説明する。まず、指定された図形を描画し、2値エッジリストを作成する。全ての図形を描画が終了すると、合成部18に2値エッジリストを出力して処理を終了し、描画していない図形がある場合には、描画以降を繰り返す。なお、上述した処理のうち、図形を描画は、多値ベクトル処理部14と同じであり、ベクトルからの2値エッジリストの作成は、多値文字処理部12における高解像度エッジリストの作成と解像度の設定、つまり仮想座標点の設定が異なるのみで、他は同じであるので、詳細な説明は省略する。図17(a)に示す図形から作成したエッジリストを表9に示す。

【表9】

※

| j | st | rl |
|---|----|----|
| 1 | 1 | 7 |
| 2 | 15 | 9 |
| 3 | 11 | 3 |

【0068】次に、グラフィクス処理部16での処理を説明する。ここでは、指定された領域を指定された色および指定された塗り潰し方法で塗り潰す処理が行われるが、塗り潰し方法が指定された領域を一様に塗り潰す方法である場合には詳述するまでもないので、一様に塗り潰さない方法について説明する。このような塗り潰し方法では、仮想座標平面上の座標位置の関数として各座標位置の画素値が決定される。この関数としては、例えば矩形領域内で主走査方向に最高濃度から最低濃度まで画

素値が単調減少する関数がある。

【0069】ここで、図18は、一様に塗り潰さない方法を説明するための概念図である。図18(a)は、主走査方向14画素、副走査方向1ラインの仮想座標平面の領域に、該領域の主走査方向の左端を最高濃度、右端を最低濃度とし、その間は主走査方向の左から右に向かって単調減少する関数により描画した例、図18

(b)は、プリンタ解像度と同じ解像度で仮想座標点を設定して各画素値を決定した例、図18(c)は、図18(b)に示す領域をプリントアウトした例である。な

お、図18(b)の上側の数値は、その直下の画素の値を示している。なお、図18に示す例は、一具体例であ*

*り、座標位置から画素値を決定する関数を様々に変えることにより、どのような領域の塗り潰し方法にも対応できるのは明らかである。ここで、座標位置によらず画素値が同じ値に決定される関数を用いれば、指定された領域を一様に塗り潰すことができるので、一様に塗り潰す方法も上述した塗り潰し方法に含まれる。画素値の決定が終了すると、決定した画素値から多値エッジリストを作成し、合成部18に出力する。表10に図18(b)に示す領域の各画素値から作成したエッジリストを示す。

【表10】

| j | st | pv | rl |
|---|----|-----|----|
| 1 | 1 | 246 | 1 |
| | 2 | 228 | 1 |
| | 3 | 210 | 1 |
| | 4 | 191 | 1 |
| | 5 | 173 | 1 |
| | 6 | 155 | 1 |
| | 7 | 137 | 1 |
| | 8 | 119 | 1 |
| | 9 | 100 | 1 |
| | 10 | 82 | 1 |
| | 11 | 64 | 1 |
| | 12 | 46 | 1 |
| | 13 | 28 | 1 |
| | 14 | 9 | 1 |

【0070】次に、イメージ処理部17での処理を説明する。ここでは、ラスターイメージデータに従って指定された領域に、指定された解像度および指定された色空間で描画するのみであるが、指定された解像度がプリンタ解像度と一致しない場合がある。このような場合には、プリンタ解像度と同じ解像度で設定した仮想座標点と、ラスターイメージデータの各画素の位置とが一致しないため、ある座標点の近傍にあるラスターイメージデータの各画素の値から近似画素値を算出することによりプリントアウトを可能にする。

【0071】つまり、解像度変換処理によって、ラスターイメージデータをプリンタ解像度に合わせることであり、プリントアウトを可能にする。このときの近似画素値の算出方法は、周知の方法でよい。また、指定された色空間がプリントアウト可能な色空間と一致しない、あるいはラスターイメージデータの各画素値が表す色がプリンタで再現できない場合には、ラスターイメージデータの画素値を変更する。このときの画素値の変更方法も周知の方法でよい。上述したように、ラスターイメージデータをプリントアウト可能な解像度および色空間の画像に変換した後、変換した画像から多値エッジリストを※50

30※作成し、合成部18に出力する。

【0072】最後に、合成部18での処理を説明する。合成部18は、多値文字処理部12、2値文字処理部13、多値ベクトル処理部14、2値ベクトル処理部15、グラフィクス処理部16およびラスターイメージ処理部17から出力されたエッジリストを合成し、ラスター画像に変換して、出力端Aを通じて外部に出力する。なお、この合成処理は、エッジリスト間の操作、例えばリストの連結・挿入・選択であり、優先順位はコマンド分離部11に入力されたPDL内での描画順序に従う。例えば、何も指定がない場合には、PDLの先頭から最後尾に向けて優先順位が高くなる。

【0073】つまり、後から描画される文字またはベクトルが先に描画した文字またはベクトルに上書きされることとなる。このとき、後から描画される文字またはベクトルのエッジリストは、既存の出力用エッジリストに強制挿入され、既存の出力用エッジリストの該当部分は書きつぶされる。また、何らかの指定があった場合には、上記描画順序に従わずに指定された描画順序に従う。例えば、PDLの先頭に記述されている文字またはベクトルを最優先で描画するよう指定があった場合に

は、後から描画される文字またはベクトルのエッジリストを既存の出力用エッジリストに合成する際に、同じ画素位置にリストのデータがなければリストを連結し、同じものがある場合には何もしない。

【0074】次に、上述した合成部18での処理の具体例を説明する。例えば、表5に示す多値エッジリストが多値文字処理部12から出力され、その後、表11に示す2値エッジリストが2値ベクトル処理部15から出力されたとする。

【表11】

| j | st | rl |
|---|----|----|
| 6 | 8 | 1 |
| 7 | 8 | 1 |
| 8 | 8 | 1 |
| 9 | 8 | 1 |

*【0075】なお、PDLの記述では、表5に示す多値エッジリストに関する記述、すなわち文字「一」を多値展開する記述が、表11に示す2値エッジリストに関する記述より先、すなわち副走査方向の画素幅「1」の直線を描画する記述より先にあるとする。ここで、描画順序に関する指定がない場合には、後から描画される文字またはベクトルが優先されるため、表12に示すようなエッジリストが作成される。つまり、表5に示す多値エッジリストにデータがない6ライン目、7ライン目および9ライン目の8番目の画素については、表11に示す2値エッジリストのデータが連結され、8ライン目の8番目の画素についてはともにデータが存在しているため、表11に示す2値エッジリストのデータが挿入されている。

【表12】

*

| j | st | pv | rl |
|---|----|-----|----|
| 6 | 8 | 255 | 1 |
| 7 | 8 | 255 | 1 |
| | 13 | 192 | 1 |
| | 14 | 64 | 1 |
| 8 | 1 | 192 | 7 |
| | 8 | 255 | 1 |
| | 9 | 192 | 3 |
| | 13 | 128 | 2 |
| 9 | 8 | 255 | 1 |

【0076】一方、文字「一」を最優先で多値展開する指定があった場合には、同じ画素位置にデータがあっても、指定された文字またはベクトルが優先されるため、表13に示すようなエッジリストが作成される。つまり、表5に示す多値エッジリストにデータがない6ライン目、7ライン目および9ライン目の8番目の画素につ

30※いては、表11に示す2値エッジリストのデータが連結され、8ライン目の8番目の画素についてはともにデータが存在しているが、指定により多値エッジリストのデータが優先されるため、リストの操作は何も行われな

【表13】

| j | st | pv | rl |
|---|----|-----|----|
| 6 | 8 | 255 | 1 |
| 7 | 8 | 255 | 1 |
| | 13 | 192 | 1 |
| | 14 | 64 | 1 |
| 8 | 1 | 192 | 12 |
| | 13 | 128 | 2 |
| 9 | 8 | 255 | 1 |

【0077】以上説明したように、合成部18では、リストの合成処理が行われ、最終的にラスター画像に変換される。ラスター画像への変換は、ランレングス形式で記述されているエッジリストをその記述内容に従ってラスター画像に変換するのみであるので、詳細な説明は省★50

★略する。また、合成部18では、多値エッジリストまたは2値エッジリストが多値文字処理部12、2値文字処理部13、多値ベクトル処理部14、2値ベクトル処理部15のいずれから出力されたかを検出し、その結果を識別データとして上記ラスター画像の画素毎に出力端B

から外部に出力する。なお、識別データ作成の際には、上述したリストの合成時の描画順序に関する処理結果が反映される。

【0078】ところで、上述したリストの合成処理の際には、同じ画素位置に、リストにデータがあるかどうかを調べるため、異なる処理部から出力されたエッジリストが表す画像の境界を容易に知ることができる。したがって、このような境界に対する処理、例えばグラフィクス領域中に黒色文字が埋め込まれているような場合に、プリント時のレジストレーションのずれを目立たなくする

ような境界処理も容易に行える。この境界処理は、以下のようにして実現できる。
【0079】まず、多値文字処理部12、2値文字処理部13、多値ベクトル処理部14、2値ベクトル処理部15のいずれかから出力されたエッジリストから画像の境界を検出する。この検出は、当該エッジリストをラスター画像に変換した場合に「0」でない画素値の画素群の主走査方向の始点と終点とを検出すること、つまりエッジリスト内で同じラインに位置する各リストデータの主走査方向の始点と終点とを調べ、さらに1つのリストデータの終点と他のリストデータの始点とが主走査方向に隣接する場合には、その2つのリストデータの隣接する始点と終点とを検出対象からはずすことにより行うことができる。

【0080】次に、合成部18において、グラフィクス処理部16またはイメージ処理部17から出力されたエッジリストと、多値文字処理部12、2値文字処理部13、多値ベクトル処理部14、2値ベクトル処理部15のいずれかから出力されたエッジリストとを合成する際には、例えばグラフィクス処理部16またはイメージ処理部17から出力されたエッジリスト中のリストデータで、検出した画像の境界と始点あるいは終点とが一致するとき、一致した点が始点であった場合には、その値をデクリメントし、終点であった場合には、インクリメントする。このように境界処理を行えば、プリント時のレジストレーションのずれを目立たなくすることができる。

【0081】上述した処理に対して、エッジリスト合成時にリストデータを変更するのではなく、境界であることをどの処理部から出力されたかを示す識別データとは異なる内容の識別データとして出力端Bから外部に出力するようにしてもよい。

【0082】また、別の境界処理として、多値文字と背景との重ね合わせ処理がある。以下では、当該重ね合わせ処理の例を図14を参照して説明する。ここで、図19(a)は、色成分をYMC Kの4成分として背景上に理想的に展開した例を示す概念図であり、各色成分の各画素は256階調である。図19(b)、(c)は、図19(a)に示した画像のうち、文字を多値展開した画像のそれぞれK成分・C成分を示す概念図、図19

(d)は、図19(b)、(c)に示す画像をプリントアウトした例を示す概念図である。なお、図19において、矩形は画素を表し、白色でない矩形の画素値は階調を有する画素であり、図中の凡例に従う。

【0083】通常、文字を白色ではない背景に重ね合わせる場合、図19(c)に示すように、背景の画素のうち文字の有色画素と同じ位置にある画素は、文字の画素に置き換えられる。これに対して、多値展開した文字は、パルス幅変調等の解像度を拡張する印字方法により印字されるため、特に画素値が中間調である画素については、図19(d)に示すように、色成分間で白隙を生じることがある。したがって、多値展開文字の中間調画素を背景に重ね合わせる場合には、背景画素を多値展開文字の画素に単純に置き換えるのではなく、背景画素の色成分のうち、多値展開文字とは異なる色成分の画素の値を保存し、同一色成分の画素のみ背景画素を文字画素に置き換えるようにすればよい。図19(e)は、図19(a)に示す画像を文字と背景の重ね合わせを上述した方法で行った場合のプリントアウト例を示す概念図である。上述した境界処理の実現方法は、プリント時のレジストレーションのずれを目立たなくする境界処理と同様であるため、詳細な説明は省略する。なお、上述において、ラスターイメージは、ラスターイメージ処理部17でエッジリストに変換された後、他の処理部から出力されたエッジリストと合成処理されるとしているが、ラスターイメージ処理部17では、ラスターデータの取り出しだけを行い、ラスターイメージ処理部17から出力されるエッジリストを合成し、ラスターライズした後、当該ラスターデータとの合成処理を行うようにしてもよい。

【0084】C. 変形例

以上、本発明の一実施形態について説明したが、本発明の実施形態は、これに限定されるわけではなく、例えば上記実施形態の合成部18においては、複数の処理部から出力されたエッジリストをラスターライズする前に合成処理を行っているが、各エッジリストをラスターライズした後に合成処理を行っても、本発明の趣旨は何ら逸脱しない。以下では、本発明の変形例として後者の実施形態について説明する。

【0085】この変形例では、上記実施形態とは、合成部18における合成処理が異なるのみで、コマンド分離部11および各処理部における処理については同じであるため説明を省略する。合成部18は、まず、各処理部から出力されたエッジリストをラスターライズする。ラスターライズした結果得られるラスター画像は、合成処理を行うために合成部18内のメモリ（図示せず）に書き込まれる。メモリへは、メモリ上のラスター画像がそのまま外部に出力できるよう書き込まれる。具体的には、書き込もうとする画素のメモリ上での位置関係が画像全体での位置関係と一致するように書き込まれる。

【0086】しかしながら、書き込みの際には、他の処理部から出力されたエッジリストをラスターライズしたラスト画像が書き込まれている場合がある。したがって、この場合には、上記実施形態と同様に、PDL内での描画順序に応じて書き込もうとする画素を、書き込もうとする位置の既存の画素に上書きするのか、上書きせずに既存の画素を保存するのかを決定する。この画素の上書きの可否決定は、上記実施形態の合成部18におけるエッジリスト間の合成処理に相当するが、エッジリストとラスター画像との違いはあるが合成論理は変わらない。したがって、出力端Aから外部に出力される画像については何ら相違はない。

【0087】また、上記実施形態の合成部18における文字と背景との境界処理についてもエッジリスト間の合成処理と同様にラスター画像間で処理を行うことができる。上記実施形態では、文字と背景との境界を検出した後、境界部分のエッジリストの内容を操作することにより、プリント時のレジストレーションのずれを防止する処理あるいは多値展開文字と背景とを重ね合わせる際の多値文字周辺の白隙を防止する処理を行っている。これに対して、本変形例では、境界を検出した後、メモリに書き込む際に上記処理を行う。具体的には、プリント時のレジストレーションのずれを防止するために、境界部分の文字の画素をメモリに書き込む前に、その画素の主走査方向の1つ前あるいは1つ後の画素値のうち、文字とは異なる色成分の画素値を書き込む文字の画素の同一色成分の画素値とする。また、多値展開文字と背景とを重ね合わせる際に多値文字周辺の白隙を防止するために、多値展開文字の中間調画素をメモリに書き込む前に、その画素の主走査方向の1つ前あるいは1つ後の画素値のうち、文字とは異なる色成分の画素値を、書き込む文字の画素の同一色成分の画素値とする。

【0088】いずれも、上記実施形態でエッジリストの内容を操作する処理を、ラスター画像間の処理に置き換えただけであり、両者とも最終的に出力端Aから出力される画像の内容に相違はない。なお、書き込む画素の文字とは異なる色成分の画素値を、当該画素の主走査方向の1つ前あるいは1つ後の画素の同一色成分の値としているが、1つ前あるいは1つ後の画素のどちらを選択するかは上記実施形態における、プリント時のレジストレーションのずれを防止する処理での選択と同一論理とする。

【0089】ところで、上述した実施形態および変形例は、いずれもPDLを出力するアプリケーションソフトウェアとプリントエンジンとの間に実装すればよいので、いくつかの異なる形態により実現することができる。例えば、印字を行うプリントエンジンと、プリントエンジンの制御、外部との通信の制御等を行うプリントコントローラとが1つの筐体に納められているプリンタでは、プリントコントローラで動作するソフトウェアと

して実現できる。また、プリントエンジンとプリントコントローラとが別個の筐体となっている場合にも、プリントコントローラで動作するソフトウェアとして実現できる。あるいは、PDLを出力するアプリケーションソフトウェアが搭載されているPCあるいはWSで動作するソフトウェアとしても実現できる。さらに、処理速度を速めるために上記実施形態における処理の一部あるいは全部をハードウェアで実現してもよく、いずれの実施形態で実現しても本発明の趣旨を何ら逸脱するものではない。

【0090】以上、本発明による画像処理装置の実施例について述べたが、本発明による画像処理装置の実現形態は、上述した内容に限定されるものではなく、他の実現形態であっても本発明の趣旨を逸脱しない範囲ならばどのような実現形態であっても構わない。

【0091】

【発明の効果】以上、説明したように、この発明によれば、ページ記述言語で表され、入力手段から入力された画像情報の中から抽出手段により文字または線画部を抽出し、抽出した文字または線画部を第1の変換手段により2値の階調を有する第1の中間コードに変換し、第1の中間コードを第2の変換手段により上記第1の中間コードより低い解像度で多値の階調を有する第2の中間コードに変換し、また、タグ生成手段により、第1の中間コードまたは第2の中間コードに基づいて、入力された画像情報の所定の領域ごとに当該領域を占める画像の種類を表し、当該第2の中間コードで表された文字または線画部であることを表す情報を少なくとも含むタグ情報を生成するようにしたので、ページ記述言語で記述された画像情報をラスター画像に変換する際に必要な記憶容量を少なくすることができ、より高画質のプリントアウトを得ることができるという利点を得られる。

【図面の簡単な説明】

【図1】 本発明による画像処理装置の一構成例を示すブロック図である。

【図2】 本発明による画像処理装置に入力されるPDLの一例を示す概念図である。

【図3】 エッジリスト生成過程を示すフローチャートである。

【図4】 2値エッジリスト作成処理の詳細を示すフローチャートである。

【図5】 2値エッジリスト作成処理の詳細を示すフローチャートである。

【図6】 2値エッジリスト作成処理の詳細を示すフローチャートである。

【図7】 2値エッジリスト作成処理の詳細を示すフローチャートである。

【図8】 先頭ライン処理の詳細を示すフローチャートである。

【図9】 先頭ライン処理の詳細を示すフローチャート

である。

【図10】 最終ライン処理の詳細を説明するためのフローチャートである。

【図11】 最終ライン処理の詳細を説明するためのフローチャートである。

【図12】 変数n1を算出するアルゴリズムを説明するためのフローチャートである。

【図13】 変数n1を算出するアルゴリズムを説明するためのフローチャートである。

【図14】 処理過程でのラスター画像ならびに比較対照のための概念図である。

【図15】 図14(a), (b)の部分拡大図である。

【図16】 多値ベクトル処理部14での処理を示すフローチャートである。

【図17】 多値ベクトル処理部14での処理の具体例

を説明するための概念図である。

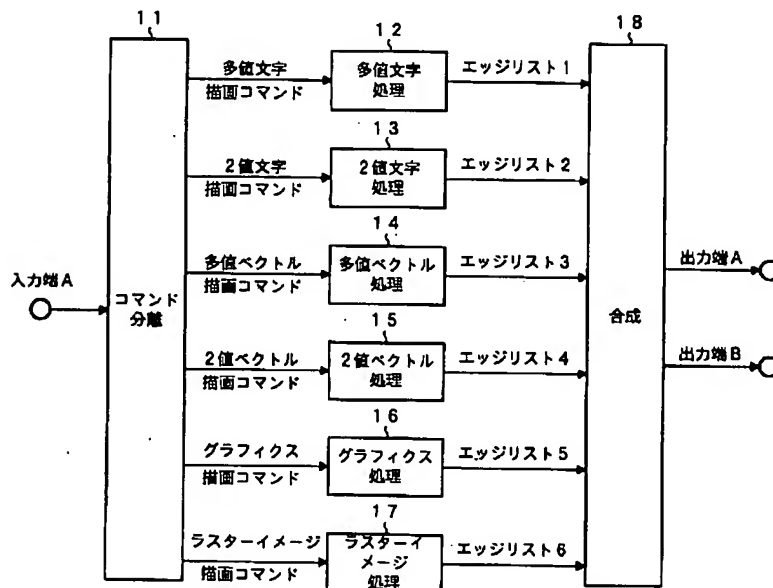
【図18】 グラフィクス処理部16での指定された領域を一様に塗り潰さない方法を説明するための概念図である。

【図19】 多値文字と背景との重ね合わせ処理を説明するための概念図である。

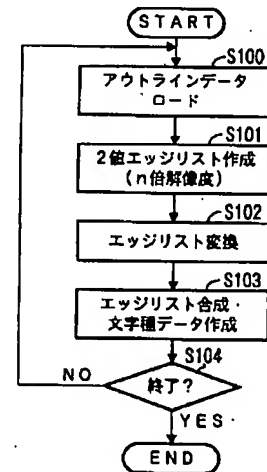
【符号の説明】

- 11 コマンド分離部（入力手段、抽出手段）
- 12 多値文字処理部（第1の変換手段）
- 13 2値文字処理部（第1の変換手段）
- 14 多値ベクトル処理部（第1の変換手段）
- 15 2値ベクトル処理部（第1の変換手段）
- 16 グラフィクス処理部
- 17 ラスターイメージ処理部
- 18 合成部（第2の変換手段、タグ生成手段、合成手段、境界検知手段）

【図1】



【図3】



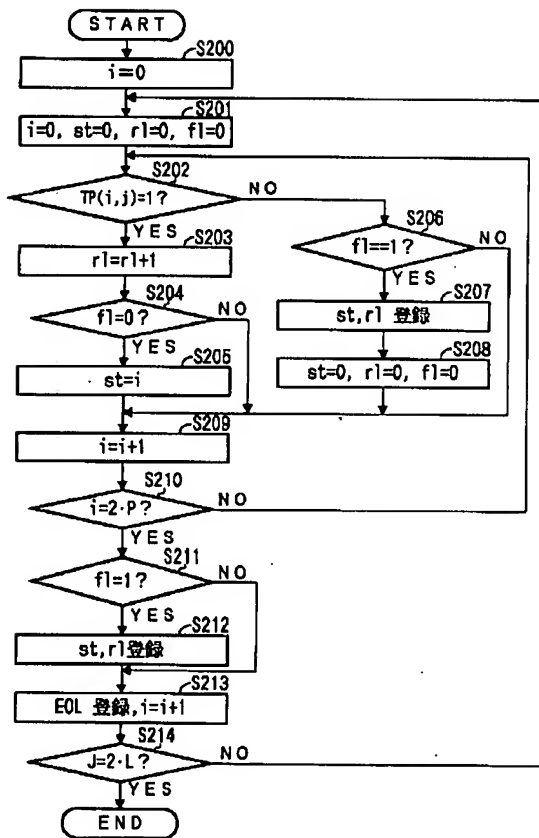
【図2】

```

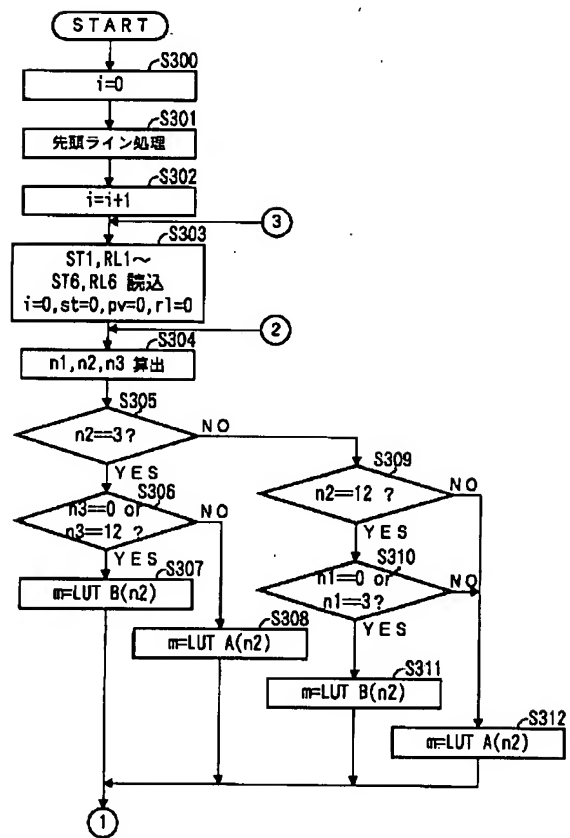
TYPE (x1, y1, Mincho, 12, abcdefg, mv)
TYPE (x1, y1, Mincho, 12, abcdefg, bn)
LINE (x2, y2, x3, y3, 1, mv)
LINE (x2, y2, x3, y3, 1, bn)
PAINT (x4, y4, x5, y5, RED, RGB)
RASTER (x6, y6, p, 1, rx, ry, RGB) [fff...fff]

```

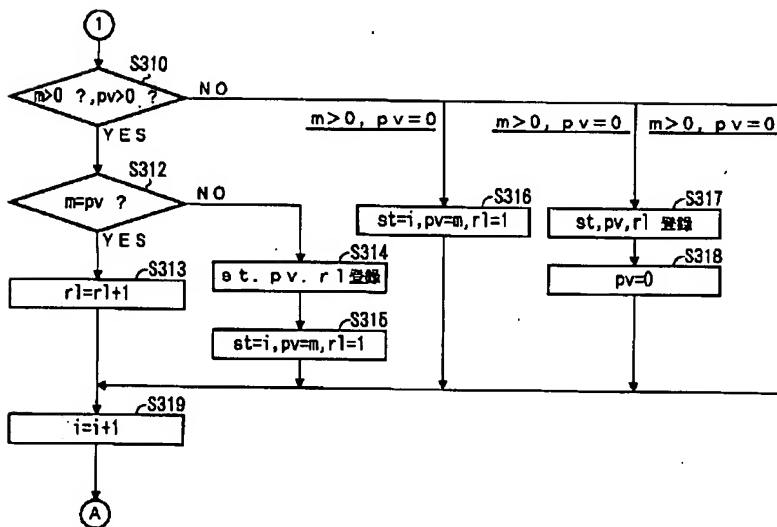
【図4】



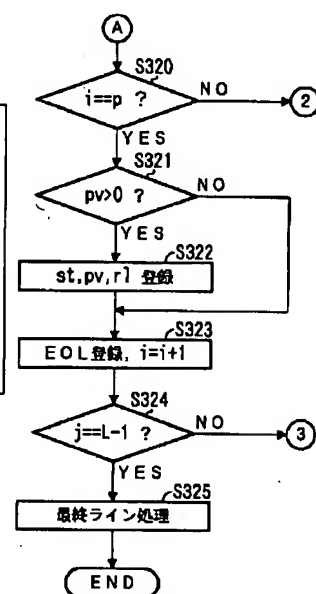
【図5】



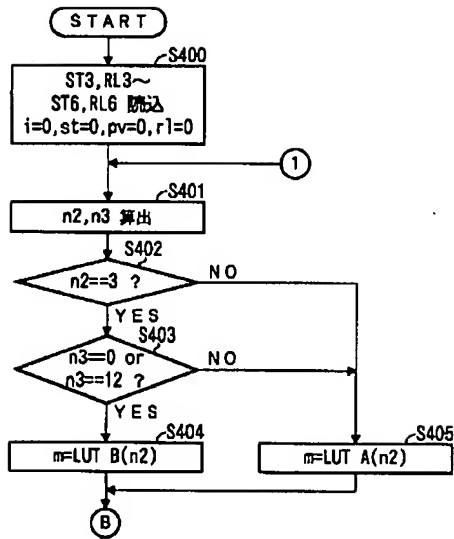
【図6】



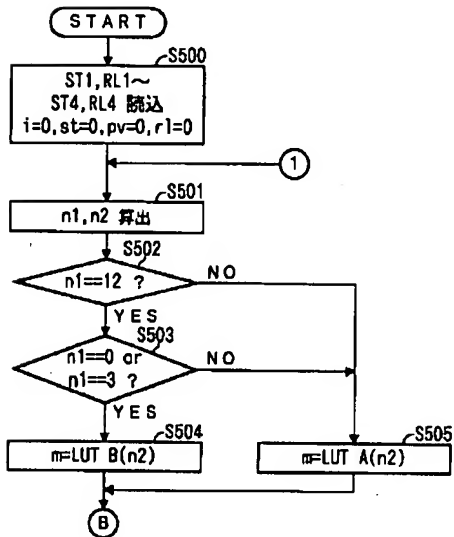
【図7】



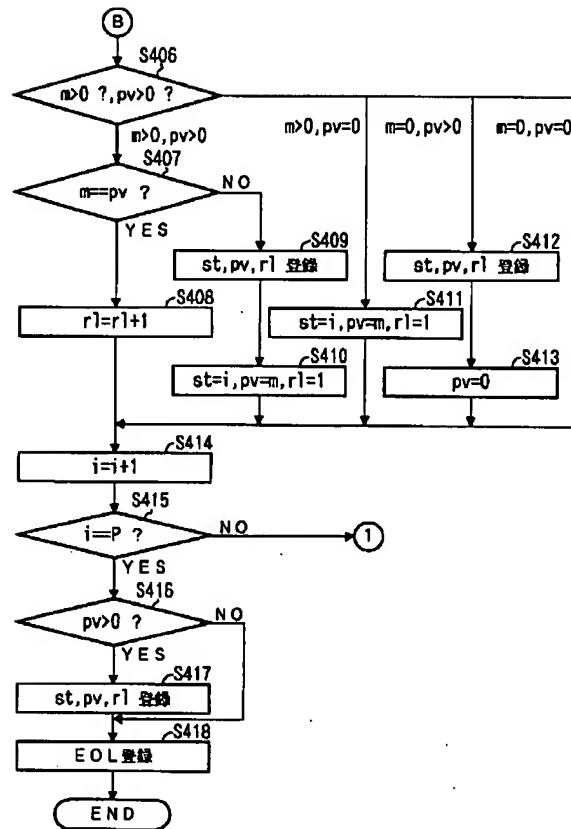
【図8】



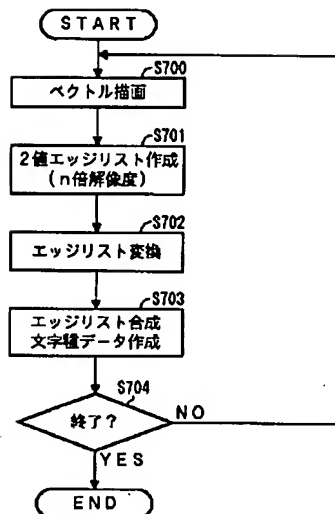
【図10】



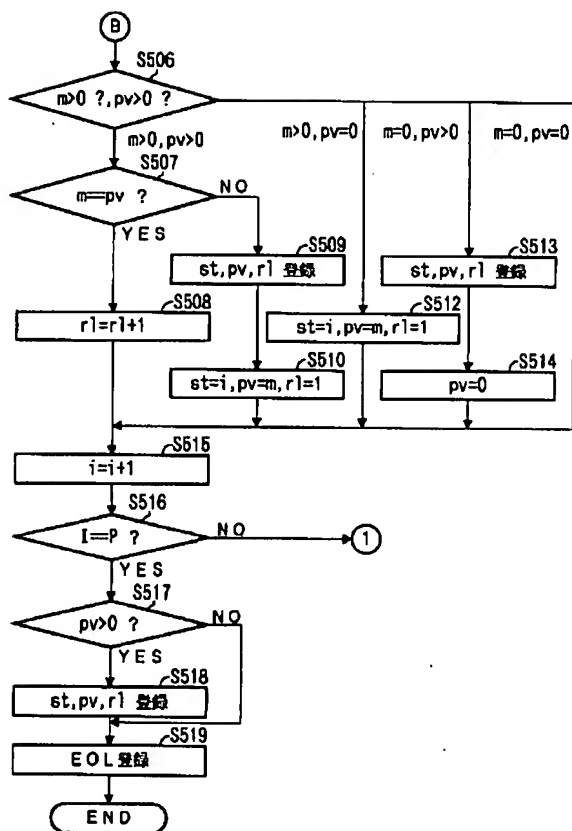
【図9】



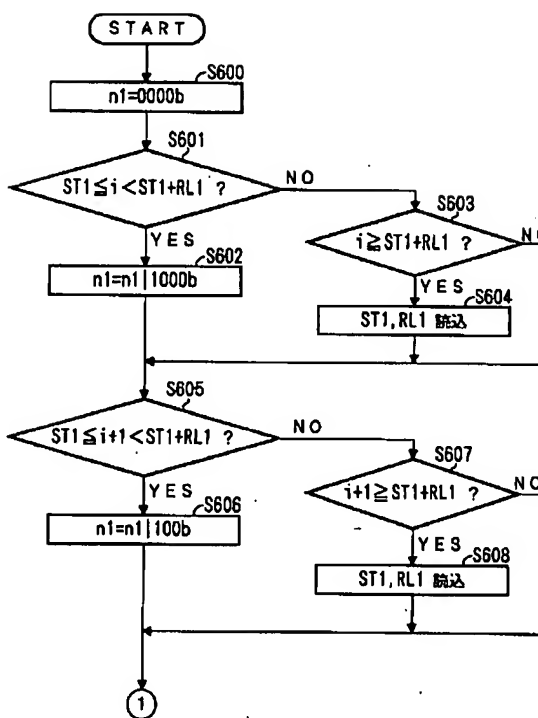
【図16】



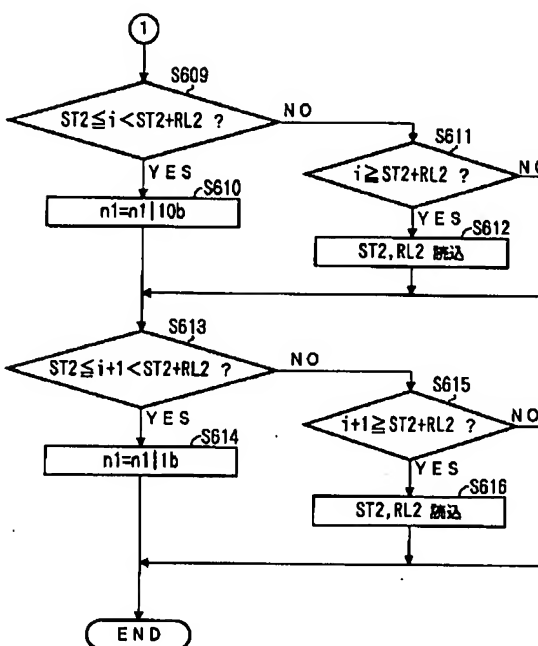
【図11】



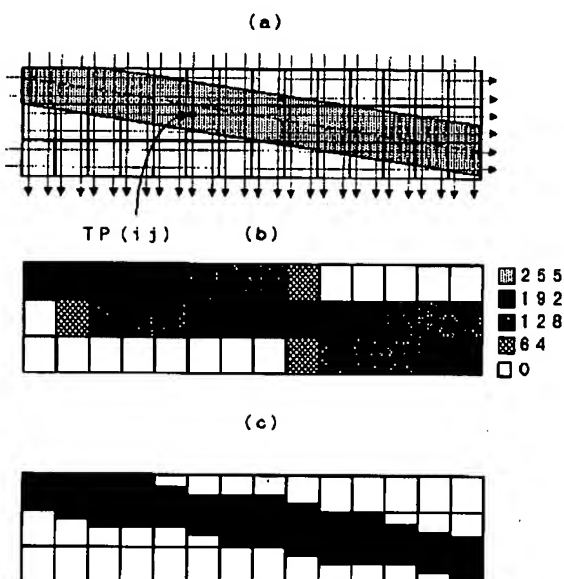
【図12】



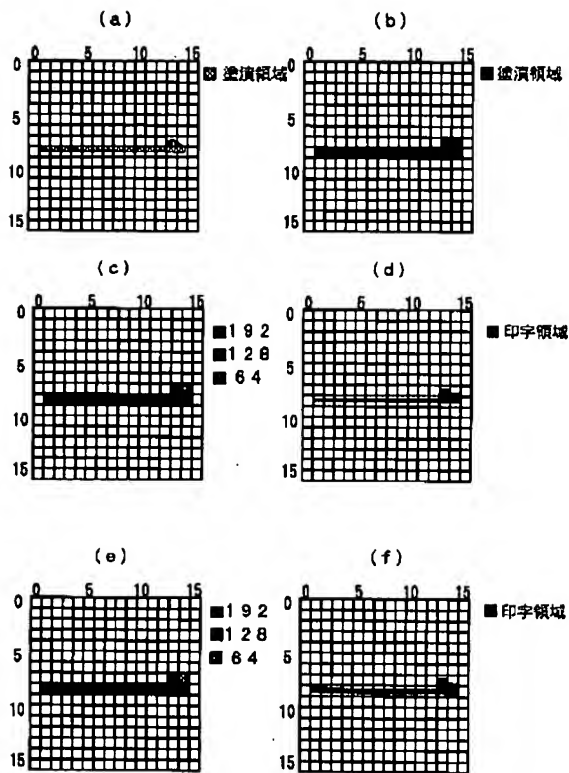
【図13】



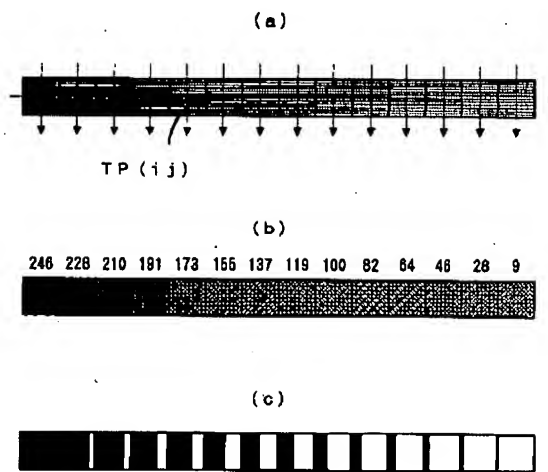
【図17】



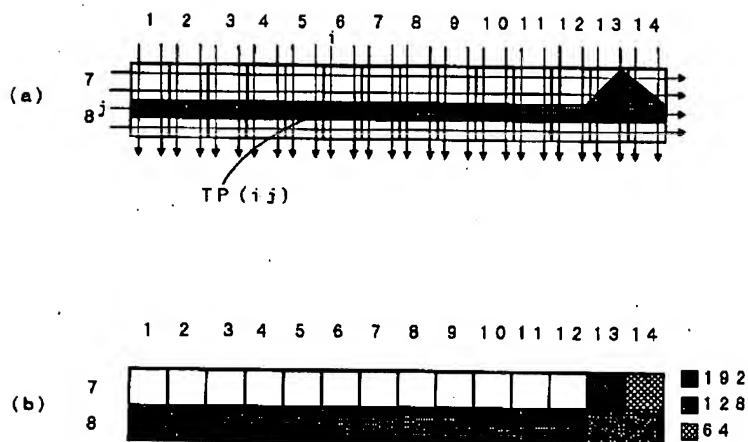
【図14】



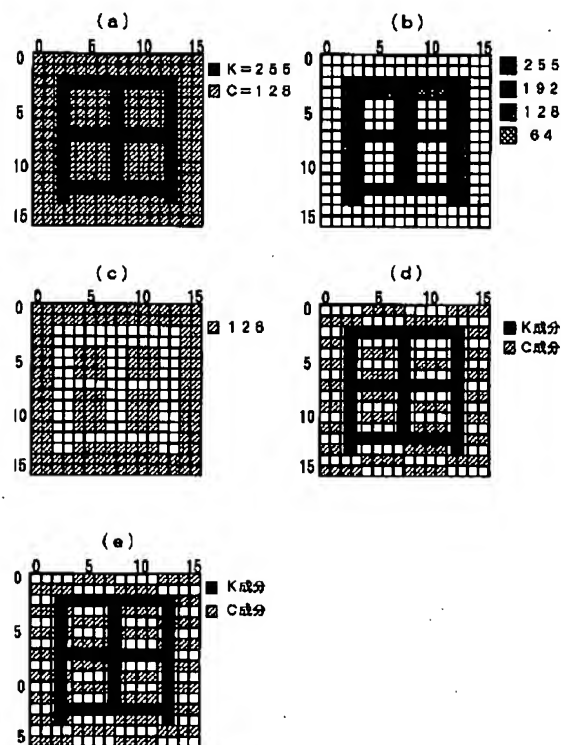
【図18】



【図15】



【図19】



フロントページの続き

(51)Int.Cl.⁷G 0 9 G 5/00
5/40

識別記号

5 2 0

F I

B 4 1 J 3/12
G 0 6 F 15/72

テーマコード(参考)

G 9 A 0 0 1
3 5 0

Fターム(参考) 2C062 AA24 AA63

2C262 AB07 AB09 AC15 AC17 DA01
DA02 DA03 DA16 DA18 EA07
5B021 AA02 BB02 CC05 CC06 EE01
5B080 EA04 FA05 FA14
5C082 AA01 BA02 BA12 BA13 BA20
BA27 BA39 CA22 CA54 CA55
CB01 DA87 MM04 MM10
9A001 HH23 HH28 JJ35 KK42 KK54